



TITLE:

あるクラスの文脈依存グラフ文法 とその性質 (計算モデルとアルゴリズム)

AUTHOR(S):

安達, 由洋; 小林, 卓; 土田, 賢省; 夜久, 竹夫

CITATION:

安達, 由洋 ...[et al]. あるクラスの文脈依存グラフ文法とその性質 (計算モデルとアルゴリズム). 数理解析研究所講究録 1999, 1093: 224-229

ISSUE DATE:

1999-04

URL:

<http://hdl.handle.net/2433/62943>

RIGHT:

あるクラスの文脈依存グラフ文法とその性質

東洋大学 安達 由洋 (Yoshihiro Adachi)
東洋大学 小林 卓 (Suguru Kobayashi)
東洋大学 土田 賢省 (Kensei Tuchida)
日本大学 夜久 竹夫 (Takeo Yaku)

1 はじめに

グラフ文法は文字列を基にした形式言語理論のグラフへの一般化であり、視覚的オブジェクトの生成・変換を数学的に厳密にかつ分かりやすくモデル化することのできるメカニズムを提供する。1960年代に始まったグラフ文法の研究は着実に発展を続け、これまでに多くの異なったタイプのグラフ文法が定式化されている ([1])。また、グラフ文法のビジュアル・システムへの実際的应用についても多くの研究が報告されている ([2, 3, 4, 5])。

文脈依存グラフ文法は、2つ以上のノードからなるグラフを左辺に持つ文脈依存プロダクションを用いることができるので、各々のプロダクションの左辺がただ1つのノードからなる文脈自由グラフ文法に比べてより柔軟に視覚的オブジェクトの構文規則を記述することができる。Mariottら [6] は、視覚的言語の表現能力を研究する中で文脈依存記述は視覚的言語にとって重要であると結論づけている。

安達ら [7, 9] は、導出過程においてプロダクションの結合命令に従い隣接ノードと埋め込むグラフのノードの間にのみ新しいエッジを張るような文脈依存グラフ文法を定義した。そして、この文法に基づいてブロック線図 [7, 8, 9, 10] やシグナルフローグラフ [11, 12] などに対する構文規則を定式化した。Rekersら [13] は、layered graph grammar と呼ぶ文脈依存グラフ文法を定義し、その構文解析法について議論している。Zhangら [14] は、この layered graph grammar をもとに reserved graph grammar を開発した。この reserved graph grammar も文脈依存グラフ文法であり、彼らはこの文法に対して多くの場合に多項式時間で構文解析できるアルゴリズムを報告している。しかしながら、これらの研究を除いては文脈依存グラフ文法についての理論的あるいは実用的な研究は少ない。

本研究では、まず NCE 文脈依存グラフ文法 (context-sensitive graph grammar with neighborhood controlled embedding, NCE-CSGG) を定義する。このグラフ文法は、我々が以前定義した文脈依

存グラフ文法 [9] を一般化したものであり、またよく知られている NCE (context-free) graph grammar [1] の文脈依存形への拡張となっている。

次に、NCE 文脈依存グラフ文法のプロダクション・コピーの合成とその結合律について議論する。プロダクション・コピーの合成は、制御システムにフィードバック結合を埋め込む場合や、プログラム図で部分図を再利用する場合のように、特定の構造を持つ部分グラフをどのように埋め込むかをグラフ書き換え規則で定義するために理論的にも実用的にも重要である。本研究は、ユーザが望む構造を埋め込むための方法をグラフ文法に基づいて表現する初めての試みである。

さらに、NCE 文脈依存グラフ文法の合流性とその条件について議論する。一般に、NCE 文脈依存グラフ文法に基づいて導出されるグラフは、プロダクション・コピーの適用の順序に依存して変わる。合流性とは、導出の結果得られるグラフがプロダクション・コピーの適用の順番に依存しない性質である。この合流性は、合成プロダクション・コピーの適用の妥当性を保証し、また効率的な構文解析アルゴリズムの開発のためにも重要な性質となる。

2 NCE 文脈依存グラフ文法

部分グラフの書き換えを用いた NCE 文脈依存グラフ文法を定義する。この文法では、グラフ書き換えにおける埋め込み過程で、隣接ノードと埋め込んだノード間にのみ接続命令に従って新しいエッジを作る。

定義 1 (グラフ)

Σ をノードラベルのアルファベット、 Γ をエッジラベルのアルファベットとする。 Σ と Γ 上の グラフ とは、3つ組 $H = (V, E, \lambda)$ である。ただし、

- (1) V は ノード の空でない有限集合である。
- (2) $E \subseteq \{(v, \gamma, w) \mid v, w \in V, v \neq w, \gamma \in \Gamma\}$ は エッジ の有限集合である。
- (3) $\lambda: V \rightarrow \Sigma$ は ノードラベリング関数 である。

グラフ H の構成要素を各々 V_H, E_H, λ_H と書くことにする。□

グラフ H と K に対して全単射 $\theta: V_H \rightarrow V_K$ が存在して, $E_K = \{(\theta(v), \gamma, \theta(w)) \mid (v, \gamma, w) \in E_H\}$, かつ, すべての $v \in V_H$ に対して $\lambda_K(\theta(v)) = \lambda_H(v)$ が成り立つとき, H と K は 同形である (isomorphic) といい, θ を H から K への 同形写像 (isomorphism) という. $V_H \cap V_K = \phi$ のとき, H と K は 互いに素である (disjoint) という. グラフ H に対して, H に同形なすべてのグラフの集合を $[H]$ で表す. このとき H を コンクリート・グラフ (concrete graph) といい, $[H]$ を アブストラクト・グラフ (abstract graph) という. 通常, コンクリート・グラフとアブストラクト・グラフを厳密には区別しない. Σ と Γ 上のすべての (コンクリート・) グラフの集合を $GR_{\Sigma, \Gamma}$ で表し, すべてのアブストラクト・グラフの集合を $[GR_{\Sigma, \Gamma}]$ で表す. $[GR_{\Sigma, \Gamma}]$ の部分集合を グラフ言語 という.

NCE 文脈依存グラフ文法が有向グラフを生成するとき, これを明示するために小文字 'd' を頭に添える. また, 生成されたグラフがノードラベルとともにエッジラベルも付いているとき小文字 'e' を頭に添える.

以下に, edNCE 文脈依存グラフ文法を定義する. NCE 文脈依存グラフ文法, dNCE 文脈依存グラフ文法, eNCE 文脈依存グラフ文法はこれの特別な場合としてみなすことができる.

定義 2 (edNCE 文脈依存グラフ文法)

edNCE 文脈依存グラフ文法とは 6 つ組

$G = (\Sigma_n, \Sigma_t, \Gamma_n, \Gamma_t, S, P)$ である. ただし,

- (1) Σ_n は 非終端ノードラベルのアルファベット である.
- (2) Σ_t は 終端ノードラベルのアルファベット である. Σ_n, Σ_t は空でない有限集合であり, 互いに素である.
 $\Sigma = \Sigma_n \cup \Sigma_t$ はノードラベルのアルファベットである.
- (3) Γ_n は 非終端エッジラベルのアルファベット である.
- (4) Γ_t は 終端エッジラベルのアルファベット である. Γ_n, Γ_t は空でない有限集合であり, 互いに素である.
 $\Gamma = \Gamma_n \cup \Gamma_t$ はエッジラベルのアルファベットである.
- (5) $S \in \Sigma_n$ はエッジのないノード 1 点からなるスタートグラフ G_s のノードのラベルである.

- (6) P は プロダクション の空でない有限集合である. ここで, プロダクションは組 $((A, X) ::= (B, Y), C)$ の形で表されるものとする. ただし,

- (a) $A \in GR_{\Sigma, \Gamma}, X \in GR_{\Sigma_n, \Gamma}$, そして X は A の誘導部分グラフである. $A \neq X$ のとき, $K = A - X$, すなわち, ノード集合 $V_{A-X} = \{v \mid v \in V_A, v \notin V_X\}$ により誘導される A の部分グラフ K を 文脈部分グラフ (context subgraph) という.
- (b) $Y \in GR_{\Sigma, \Gamma}$, かつ $1 \leq |V_X| \leq |V_Y|$ である. $B \in GR_{\Sigma, \Gamma}$ は, A, X, Y および C に依存して決まるグラフである. (依存の仕方については定義 3 のあとで具体的に説明する.)
- (c) $C \subseteq \Sigma \times V_X \times \Gamma \times \Gamma \times V_Y \times \{\text{in}, \text{out}\}$ は 結合関係 (connection relation) である.

プロダクション $p = ((A, X) ::= (B, Y), C)$ において, (A, X) を p の 左辺 と呼び $lhs(p)$ と書く. また, (B, Y) を p の 右辺 と呼び $rhs(p)$ と書く. C の各要素 $(\sigma, v_x, \beta, \gamma, v_y, d)$ を p の 結合命令 (connection instruction) という. \square

2 つのプロダクション $p_1 = ((A_1, X_1) ::= (B_1, Y_1), C_1)$ と $p_2 = ((A_2, X_2) ::= (B_2, Y_2), C_2)$ に対して, A_1 から A_2 への (左辺の) 同形写像 θ_l と B_1 から B_2 への (右辺の) 同形写像 θ_r が存在して $C_2 = \{(X, \theta_l(v_x), \beta, \gamma, \theta_r(v_y), d) \mid (X, v_x, \beta, \gamma, v_y, d) \in C_1\}$ が成り立つとき, p_1 と p_2 は 同形である という. また, 写像の対 (θ_l, θ_r) を p_1 から p_2 への 同形写像 という. G のプロダクション集合 P には, 同形なプロダクションを重複して含まないと仮定する. $p \in P$ に同形なプロダクションの集合を $copy(p)$ で表す. $copy(p)$ の要素を G の p に対する プロダクション・コピー あるいは単に プロダクション・コピー という. $copy(P) = \cup_{p \in P} copy(p)$ とする.

edNCE 文脈依存グラフ文法に基づくグラフの書き換えは次のように行われる. 与えられた ホスト・グラフ (host graph) を H とし, A' を H の誘導部分グラフとする. また, $p' = ((A', X') ::= (B', Y'), C')$ をある $p \in P$ のプロダクション・コピーとする. ただし, Y' と H が互いに素である. グラフ書き換えは以下の手順で行われる. (1) H から マザー・グラフ (mother graph) X' と X' のノ

ドに接続している H のエッジをすべて取り除き、得られたレスト・グラフ (rest graph) を H^- とする。(2) H^- にドーター・グラフ (daughter graph) Y' を置く。(3) C' の結合命令にしたがって Y' のノードと H^- のノード間のエッジを新しく作る。これを Y' の H^- への埋め込み (embedding) という。

埋め込み過程において、結合命令 $(\sigma, x, \beta, \gamma, y, in)$ $\in C'$ は次の意味を持つ。ホスト・グラフ H においてラベル σ を持つノード $v \in V_H - V_{X'}$ から $x \in V_{X'}$ へ向かうエッジ (v, β, x) があれば、 v から $y \in V_{Y'}$ へ向かうエッジ (v, γ, y) を新しく作る。このとき、マザー・グラフのノードとの間にエッジを持つレスト・グラフのノード、例えば v を隣接ノードという。結合命令 $(\sigma, x, \beta, \gamma, y, out) \in C'$ の場合も、エッジの向きが異なるだけであとは同様である。埋め込み過程でドーター・グラフとレスト・グラフの間に新しく作られるエッジはブリッジと呼ばれることがある。NCE 文脈依存グラフ文法でのグラフ書き換えはグラフの局所的な変換である。また、端点にノードの対応しない垂れ下がったエッジ (dangling edge) は生成されない。

以上に説明した edNCE 文脈依存グラフ文法のグラフ書き換えを形式的に定義すると次のようになる。
定義 3 (導出) $G = (\Sigma_n, \Sigma_t, \Gamma_n, \Gamma_t, S, P)$ を edNCE 文脈依存グラフ文法とする。 $H \in GR_{\Sigma, \Gamma}$, A' を H の誘導部分グラフとする。 $p \in P$ を G のプロダクションとする。そして $p' = ((A', X') ::= (B', Y'), C')$ は $p \in P$ に対するプロダクション・コピーとする。ただし、 H と Y' は互いに素である。このとき、プロダクション・コピー p' のグラフ H への左適用とはグラフ $H' \in GR_{\Sigma, \Gamma}$ を以下のように構成することである。

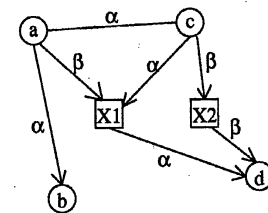
$$\begin{aligned} V_{H'} &= (V_H - V_{X'}) \cup V_{Y'}, \\ E_{H'} &= \{(v, \gamma, w) \in E_H \mid v, w \in V_H - V_{X'}\} \cup E_{Y'} \cup \{(v, \gamma, y) \mid \exists v \in V_H - V_{X'}, \exists \beta \in \Gamma, (v, \beta, x) \in E_H, (\sigma, x, \beta, \gamma, y, in) \in C', \lambda_H(v) = \sigma\} \\ &\quad \cup \{(y, \gamma, v) \mid \exists v \in V_H - V_{X'}, \exists \beta \in \Gamma, (x, \beta, v) \in E_H, (\sigma, x, \beta, \gamma, y, out) \in C', \lambda_H(v) = \sigma\}, \\ \lambda_{H'}(x) &= \begin{cases} \lambda_H(x), & x \in V_H - V_{X'} \\ \lambda_{Y'}(x), & x \in V_{Y'}. \end{cases} \end{aligned}$$

グラフ H にプロダクション・コピー p' を左適用して H' が得られるとき $H \xRightarrow{p'} H'$ と書き、 H から H'

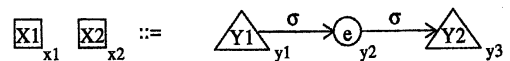
への導出ステップ (derivation step) と呼ぶ。また、導出ステップの系列を導出 (derivation) と呼ぶ。□

上記のように、プロダクション $p = ((A, X) ::= (B, Y), C)$ の左適用による導出ステップはグラフ A , X , Y と、結合関係 C により完全に決定される。グラフ B は、 $A \xRightarrow{p} B$ を満たすグラフである。したがって、 $K = B - Y = A - X$ が成り立つ。 B を用いなくてもプロダクションの記述は理論的には可能であるが、プロダクションの右辺に B を記す方がより自然であり、また文脈依存部分グラフに対応する部分グラフを把握しやすくなる。

導出ステップを例を用いて説明する。Fig. 1 (a) はホスト・グラフ H , Fig. 1 (b) はプロダクション・コピー p' である。ホスト・グラフ H にプロダクション・コピー p' を左適用して行なう導出ステップを Fig. 2 に示す。レスト・グラフ H^- が Fig. 2 (a) である。レスト・グラフ H^- にドーター・グラフを置いたのが Fig. 2 (b) である。そして、埋め込みを行ない得られた結果のグラフ H' が Fig. 2 (c) である。



(a) A Host graph H



$$C = \{(a, x1, \beta, \gamma, y1, in), (c, x1, \alpha, \sigma, y1, in), (c, x2, \beta, \sigma, y3, in), (d, x2, \beta, \gamma, y3, out)\}$$

(b) A production copy p'

Fig. 1. A host graph and a production copy

edNCE 文脈依存グラフ文法 $G = (\Sigma_n, \Sigma_t, \Gamma_n, \Gamma_t, S, P)$ に対して、導出 $H_0 \xRightarrow{p_1} H_1 \xRightarrow{p_2} \dots \xRightarrow{p_n} H_n, n \geq 0$, は $rhs(p'_i) = (B'_i, Y'_i)$ について H_0 と $Y'_i, 1 \leq i \leq n$, が互いに素であるとき 生成的である (creative) という。今後、生成的な導出のみを扱うことにする。上記のような生成的導出が $H_0 = H$ と $H_n = H'$ について存在するとき $H \xRightarrow{*} H'$ と書く。スタートグラフ G_S に対して $G_S \xRightarrow{*} H$ であるグラフ $H \in GR_{\Sigma, \Gamma}$ を G の 文形式 という。スタートグラフ G_S から導出

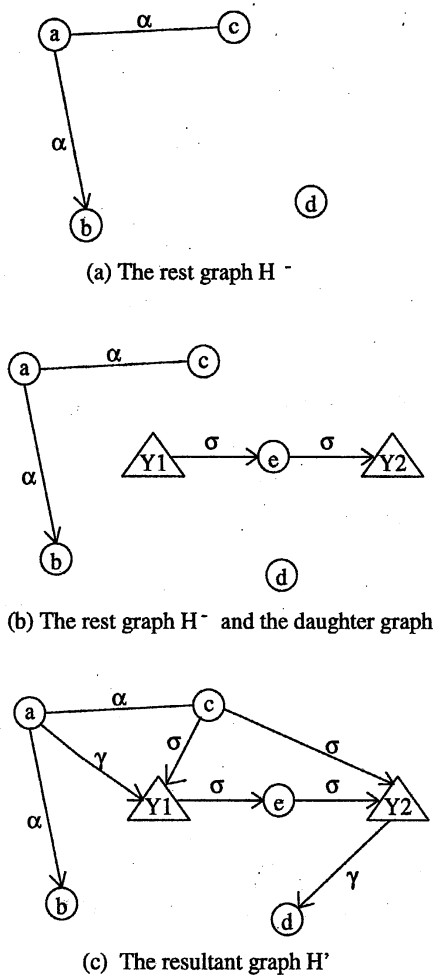


Fig. 2. The process of the derivation step

される終端グラフの集合, すなわち $\mathcal{L}(G) = \{[H] \in GR_{\Sigma, \Gamma} \mid G_S \xRightarrow{*} H \text{ for some start graph } G_S\}$ を G によって生成されるグラフ言語という.

3 プロダクション・コピーの合成

edNCE 文脈依存グラフ文法のプロダクション・コピーの合成表現は, ホスト・グラフに望みの構造を持つ部分グラフを埋め込むためのグラフ書き換え規則を定式化する理論的かつ実用的な方法である. それゆえに, プログラム図へある特定の部分プログラムを埋め込んだり, ブロック線図へフィードバック構造を埋め込むための知識を表現するのに役立つ.

定義 4 (プロダクション・コピーの合成)

$G = (\Sigma_n, \Sigma_t, \Gamma_n, \Gamma_t, S, P)$ を edNCE 文脈依存グラフ文法とし, $p_1 = ((A_1, X_1) ::= (B_1, Y_1), C_1)$ と $p_2 = ((A_2, X_2) ::= (B_2, Y_2), C_2)$ を G のプロダクション・コピーとする. このとき, A_1 と Y_2 が互いに素であり, $A_2 \subseteq B_1$ かつ $X_2 \subseteq Y_1$ であると

き, (結合関係の付随した) グラフ書き換え規則 $p = ((A_1, X_1) ::= (B, Y), C)$ を以下のように構成する.

$$B_1 \xRightarrow[p_2]{p_1} B$$

$$Y = \begin{cases} B, & (A_1 = X_1) \\ B - K_1, & \text{ここで, } K_1 = A_1 - X_1 \\ & (A_1 \neq X_1) \end{cases}$$

$$C = \{(\sigma, v, \beta, \gamma, w, d) \in C_1 \mid w \in V_{Y_1} - V_{X_2}\} \cup \{(\sigma, x, \beta, \delta, y, d) \mid \exists w \in V_{X_2}, \exists \gamma \in \Gamma, (\sigma, x, \beta, \gamma, w, d) \in C_1, (\sigma, w, \gamma, \delta, y, d) \in C_2\}$$

このとき, p を p_1 と p_2 の 合成プロダクション・コピー と呼び, $p_1 \circ p_2$ と書く. また, A_1 と Y_2 が互いに素であり, $A_2 \subseteq B_1$ かつ $X_2 \subseteq Y_1$ であるという条件が成り立つとき, $p_1 \circ p_2$ が定義可能である という. \square

合成プロダクション・コピーは, edNCE 文脈依存グラフ文法のプロダクション・コピーとまったく同じ形式のグラフ書き換え規則として表現されている. したがって, 合成プロダクション・コピーのグラフ H への左適用および導出ステップを定義 3 と同様に定義できる. また, (合成) プロダクション・コピーと (合成) プロダクション・コピーの合成も定義 4 と同様に定義される.

命題 1 $G = (\Sigma_n, \Sigma_t, \Gamma_n, \Gamma_t, S, P)$ を edNCE 文脈依存グラフ文法とする. $p_1 = ((A_1, X_1) ::= (B_1, Y_1), C_1)$ と $p_2 = ((A_2, X_2) ::= (B_2, Y_2), C_2)$ が G の (合成) プロダクション・コピーであり, $p_1 \circ p_2$ が定義可能であるとする. 任意の $H \in G_{\Sigma, \Gamma}$ に対して $H \xRightarrow[p_1]{p_1} H_1 \xRightarrow[p_2]{p_2} H_2$ かつ $H \xRightarrow[p_1 \circ p_2]{p_1 \circ p_2} H_3$ ならば $H_2 = H_3$ である. \square

ここで, ブロック線図文法 [10] を用いてプロダクション・コピーの合成例を示す. Fig. 3 にブロック線図文法のプロダクションの一部を示す. また, Fig. 4 にプロダクション・コピーの合成過程を示す. Fig. 4(d) で得られた合成プロダクション・コピー $((p_{B2'} \circ p_{B2'}) \circ p_{B10'}) \circ p_{B3'}$ は, 'Elem' とラベル付けされた非終端ノードをフィードバック構造に書き換えるグラフ書き換え規則である. この規則を左適用することにより, Fig. 5 に示すようにブロック線図にフィードバック構造を埋め込むことができる.

我々はブロック線図文法に基づいてブロック線図の編集, 構文解析, 構造認識などを支援するブロック線図処理システムを実現した. このシステムの実行画面を Fig. 6 に示す.

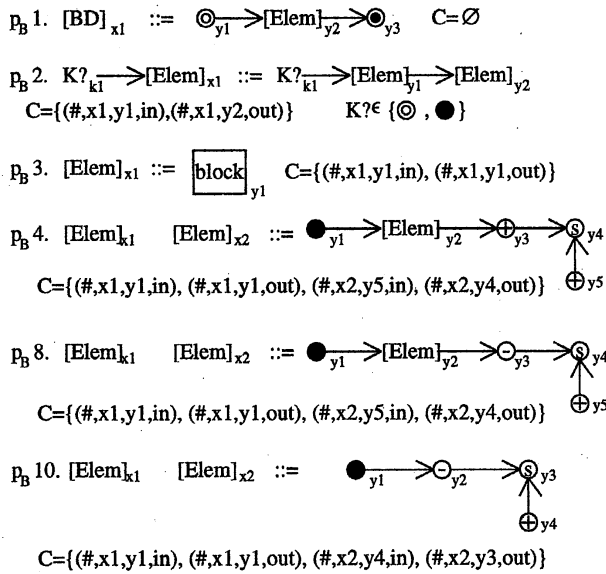


Fig. 3. Part of the productions of the block diagram grammar

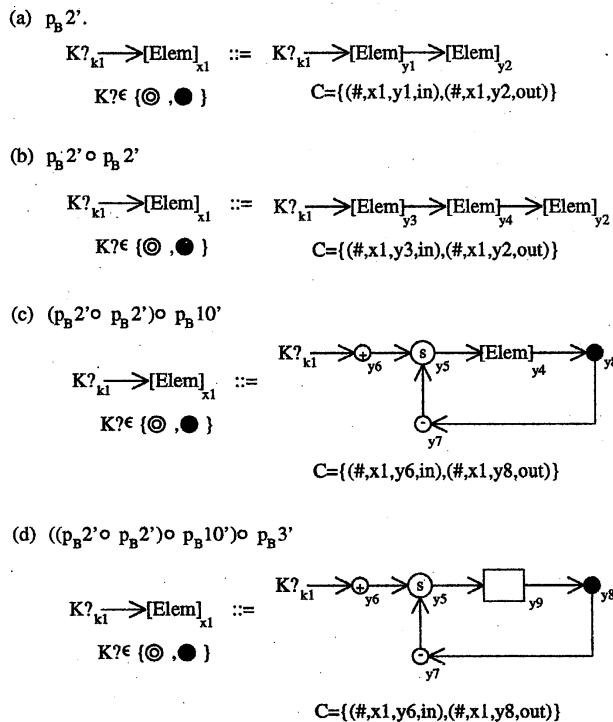


Fig. 4. An example of composite production copies

命題 2 (結合律) $G = (\Sigma_n, \Sigma_t, \Gamma_n, \Gamma_t, S, P)$ を edNCE 文脈依存グラフ文法とし, $p_1 = ((A_1, X_1) ::= (B_1, Y_1), C_1)$, $p_2 = ((A_2, X_2) ::= (B_2, Y_2), C_2)$ および $p_3 = ((A_3, X_3) ::= (B_3, Y_3), C_3)$ を G の (合成) プロダクション・コピーとする. $p_1 \circ p_2$ が定義可能であり, かつ $p_2 \circ p_3$ が定義可能であるとき,

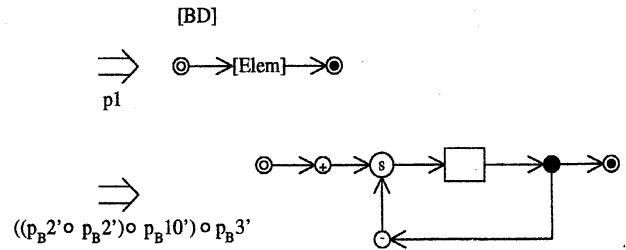


Fig. 5. Derivation using a composite production copy

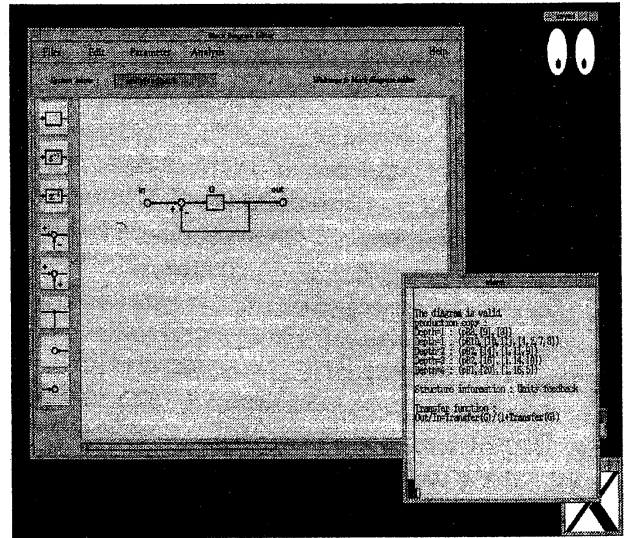


Fig. 6. A screen of our block diagram processing system

$(p_1 \circ p_2) \circ p_3 = p_1 \circ (p_2 \circ p_3)$ となる. \square

$p_1 \circ p_2$ が定義可能であり, かつ $(p_1 \circ p_2) \circ p_3$ が定義可能であっても, $p_2 \circ p_3$ が定義可能であるとは限らないので注意を要する.

4 合流性

edNCE 文脈依存グラフ文法では, (生成的) 導出の結果得られるグラフは一般にはプロダクション・コピーを左適用する順序に依存して変わる.

合流性とは, 導出の結果がプロダクション・コピーの左適用の順序に依存しないという性質のことである. この合流性は, 合成プロダクション・コピーを適用して部分グラフを埋め込んだ結果得られるグラフの文法的正しさを保証するために重要である. また, NCE 文脈依存グラフ文法に基づく効率的な構文解析アルゴリズムを開発するためにも重要な性質である.

まず, 結合関係から容易に成立するか否かを調べることのできる 静的合流性 を定義する.

定義 5 (静的合流性)

edNCE 文脈依存グラフ文法 $G = (\Sigma_n, \Sigma_t, \Gamma_n, \Gamma_t, S, P)$ は、任意の $H \in GR_{\Sigma, \Gamma}$ について $H \Rightarrow_{p'_1} H_1 \Rightarrow_{p'_2} H_{12}$ と $H \Rightarrow_{p'_2} H_2 \Rightarrow_{p'_1} H_{21}$ が G の (生成的) 導出ならば $H_{12} = H_{21}$ となるときの静的に合流的である (confluent) という。 \square

命題 3

edNCE 文脈依存グラフ文法 $G = (\Sigma_n, \Sigma_t, \Gamma_n, \Gamma_t, S, P)$ が静的に合流的である必要十分条件は、任意のプロダクション $p_1 = ((A_1, X_1) ::= (B_1, Y_1), C_1)$, $p_2 = ((A_2, X_2) ::= (B_2, Y_2), C_2) \in P$, 任意のノード $x_1 \in X_1$, $y_1 \in Y_1$, $x_2 \in X_2$, $y_2 \in Y_2$, 任意のエッジラベル $\alpha, \delta \in \Gamma$ に対して以下が成立することである。

$$\begin{aligned} \exists \beta \in \Gamma, (\lambda_{X_2}(x_2), x_1, \alpha, \beta, y_1, in) \in C_1 \text{ かつ} \\ (\lambda_{Y_1}(y_1), x_2, \beta, \delta, y_2, out) \in C_2 \\ \updownarrow \\ \exists \gamma \in \Gamma, (\lambda_{X_1}(x_1), x_2, \alpha, \gamma, y_2, out) \in C_2 \text{ かつ} \\ (\lambda_{Y_2}(y_2), x_1, \gamma, \delta, y_1, in) \in C_1 \end{aligned}$$

\square

我々が定義したブロック線図文法とシグナルフローグラフ文法は、生成されるグラフはエッジラベルを持たず、また埋め込み過程が隣接ノードのラベルに依存しないので静的に合流的である。

次に、グラフ文法分野で合流性の定義として普通用いられる動的合流性を定義する。

定義 6 (動的合流性)

edNCE 文脈依存グラフ文法 $G = (\Sigma_n, \Sigma_t, \Gamma_n, \Gamma_t, S, P)$ は、 G の任意の文形式 H について $H \Rightarrow_{p'_1} H_1 \Rightarrow_{p'_2} H_{12}$ と $H \Rightarrow_{p'_2} H_2 \Rightarrow_{p'_1} H_{21}$ が G の (生成的) 導出ならば $H_{12} = H_{21}$ となるときの動的に合流的である (dynamically confluent) という。 \square

命題 4

NCE 文脈依存グラフ文法 $G = (\Sigma_n, \Sigma_t, \Gamma_n, \Gamma_t, S, P)$ は静的に合流的であれば、動的に合流的である。 \square

5 おわりに

非常に一般的な埋め込みメカニズムを持つ NCE 文脈依存グラフ文法を定義した。次に、この文法に対してプロダクション・コピーの合成と合流性について議論した。本論文で議論した NCE 文脈依存グラフ文法は、ブロック線図 [9, 10] とシグナルフローグラフ [11, 12] のグラフ構文規則を定式化するために実際に用いられ、その理論的および実用的有効性が示されて

いる。また、合成プロダクション・コピーに基づいた部分グラフの再利用は、グラフ文法に基づいた視覚的デザイン言語へ幅広く応用することができる。

参考文献

- [1] Rozenberg, G. (Ed.), Handbook of Graph Grammars and Computing by Graph Transformation, World Scientific Publishing (1997).
- [2] Minas, M., Diagram Editing with Hypergraph Parser Support, *Proc. IEEE Symp. Visual Languages* (1997), 230-237.
- [3] Bardohl, R., GENGED A Generic Graphical Editor for Visual Languages based on Algebraic Graph Grammars, *Proc. IEEE Symp. Visual Languages* (1998), 48-55.
- [4] Zhang, D., and Zhang, K., VisPro: A Visual Language Generation Toolset, *Proc. IEEE Symp. Visual Languages* (1998), 195-202.
- [5] Münch, M., Schürr, A., and Winter A., Integrity Constraints in the multi-paradigm language PROGRES, *Proc. IEEE Symp. Visual Languages* (1998), 84-85.
- [6] Marriott, K., and Meyer, B., On the Classification of Visual Languages by Grammar Hierarchies, *Journal of Visual Languages and Computing*, 8 (1998), 375-402.
- [7] Adachi, Y., Kobayashi, S., Anzai, K., and Tsuchida, K., Block Diagram Grammar and Structure Recognition Based on Graph Rewriting, *Proc. IFAC/IEEE CACSD'97* (1997), 257-262.
- [8] Anzai, K., Adachi, Y., Kobayashi, S., and Tsuchida, K., Block Diagram Generation and Parsing Based on Graph Grammar, *Proc. IEEE ISCAS'97* (1997), 1970-1973.
- [9] Adachi, Y., Kobayashi, S., and Tsuchida, K., Multi-input/multi-output Block Diagram Grammar, *Proc. IEEE ISCAS'98 VI* (1998), 183-186.
- [10] Kobayashi, S., Adachi, Y., Tsuchida, K., and Yaku, T., Attribute Block Diagram Grammar and Its Application, *IFAC World Congress'99* (1999), (to appear).
- [11] 小林, 安達, 夜久, シグナルフローグラフのためのグラフ文法, 信学技法, COMP98-22, (1998), 9-16.
- [12] 中島, 小林, 安達, 属性シグナルフローグラフ文法, 第 57 回情報処理学会全国大会講演論文集 (1) (1998), 274-275.
- [13] Rekers, J., and Schürr, A., Defining and Parsing Visual Languages with Layered Graph Grammars, *Journal of Visual Languages and Computing* (1996), 27-55.
- [14] Zhang, D., and Zhang, K., Reversed Graph Grammar: A Specification Tool for Diagrammatic VPLs, *Proc. IEEE Symp. Visual Languages* (1997), 284-291.